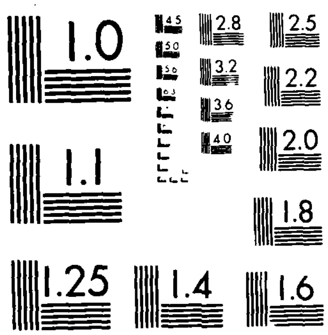
AD-A195 690 OPTICAL SYMBOLIC COMPUTING(U) COLORADO UNIV AT BOULDER 1/1
W T CATHEY ET AL. 30 APR 88 CU-1536935 AFOSR-TR-88-0661
AFOSR-86-0189

UNCLASSIFIED F/G 20/6 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A195 690

(2)

REPORT DOCUMENTATION PAGE

DTIC FILE COPY

DTIC ELECTE JUN 2 9 1988 S H D

1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) 1536935	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 88-0661
6a. NAME OF PERFORMING ORGANIZATION Optoelectronic Computing Systems Center / 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research
6c. ADDRESS (City, State and ZIP Code) Campus Box 425 Boulder, CO 80309-0425	7b. ADDRESS (City, State and ZIP Code) Building 410 Bolling Air Force Base Washington, D.C. 20332-6448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR / 8b. OFFICE SYMBOL (If applicable) NE	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-86-0189
8c. ADDRESS (City, State and ZIP Code) Bldg 410 Bolling AFB, DC 20332	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO.: 61102F; PROJECT NO.: 2305; TASK NO.: B1; WORK UNIT NO.:
11. TITLE (Include Security Classification) Optical Symbolic Computing	

12. PERSONAL AUTHOR(S)
Dr. W. Thomas Cathey, Rodney A. Schmidt

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Annual	FROM 5/87 TO 4/88	April 30, 1988	63

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report describes the results of the work on optical symbolic computing that was performed on grant AFOSR-86-0189 during the second year of the three year program. The detail of the research is in the appended papers. This section of the report is to summarize the motivation for the work, the reasons for the approach taken, and the results. First, a rationale is given for the use of optics in A.I., and then the approach taken is described. The results of the research are then given.

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☑ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Giles	22b. TELEPHONE NUMBER (Include Area Code) (202) 767-4932 / 22c. OFFICE SYMBOL NE

DD FORM 1473, 83 APR   EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

AFOSR-TR- 88-0661

ANNUAL REPORT, APRIL 30, 1988
AFOSR GRANT AFOSR-86-0189

OPTICAL SYMBOLIC COMPUTING

W. Thomas Cathey
Garret R. Moddel

Center for Optoelectronic Computing Systems
and
Department of Electrical & Computer Engineering
University of Colorado
Boulder, Colorado 80305-0425

Rodney A. Schmidt

Center for Optoelectronic Computing Systems
and
Department of Mathematics
Colorado School of Mines
Golden, Colorado 80401

The objective of the program is to design and prototype an optical inference engine and demonstrate the solution of small reasoning problem using mathematical resolution.

88 6 29 046

INTRODUCTION

This report describes the results of the work on optical symbolic computing that was performed on grant AFOSR-86-0189 during the second year of the three year program. The detail of the research is in the appended papers. This section of the report is to summarize the motivation for the work, the reasons for the approach taken, and the results.

First, a rationale is given for the use of optics in A.I., and then the approach taken is described. The results of the research are then given.

MOTIVATION

Many aerospace programs (e.g. the DARPA/Air Force Pilot's Associate Program) are making use of knowledge-based system technology, but real-time use of this technology is hampered by inadequate computing resources. However, most aspects of the technology are highly parallel and are therefore suitable for optical implementation.

Several areas of artificial intelligence, notably knowledge-based systems, pattern recognition and neural networks require so much computer power that their usefulness as problem-solving techniques are limited by present computer hardware. Our ability to address the computing power limitations of conventional processors is limited by electronic problems in the distribution of data and control signals. These problems are not present in optical computing systems. However, conventional serial approaches to A.I. problems do not map gracefully onto optical processors. A new approach is needed.

Our goal is to develop an optical data representation; design, simulate and implement basic optical operations; design, simulate and implement optical storage and data handling modules; design, simulate and implement data-dependent control; and simulate and implement a prototype optical inference engine. So far, we have developed the notation, algorithm, and logic for optical resolution; designed the resolution system using medium scale optical building blocks; generated optical and functional simulations; designed, simulated and built prototype components; and performed a functional simulation of the system.

RATIONALE FOR OPTICS

DTIC
COPY
INSPECTED
6

This project investigates the use of optics in implementing a particular approach to knowledge-based systems - the approach known as mathematical resolution. This combination of method and technology must compete with other technologies, for example VLSI, and with other methods, such as back chaining (i.e. PROLOG) and production rules. Our purpose here is to compare these approaches and show why mathematical resolution using optics is a viable and attractive approach.

The underlying thread in all knowledge-based systems is that they contain a large amount of unstructured information. By "unstructured", we mean that the relationship between different pieces of information is not strongly hierarchal. Thus the process of computation involves a relatively unselective examination of combinations of data to find sequences of information within the data which satisfy the requirements of logic and solve the problem at hand.

For
I
ion/
lity Codes
Avail and/or
Dist Special
A-1

If the data space is large, many subsequences of information exist within the system which can be independently discovered. This offers the possibility of increasing the speed of a knowledge-based system by the use of parallelism, either electronic or optical. However, the search for subsequences must have overall coordination so that all meaningful possibilities are examined, but that the same possibilities are not redundantly examined.

There is a tradeoff between the size of the data space and the complexity of the algorithms required to manipulate it. In the simplest notation, the propositional calculus, each entity in the problem space must be separately described in the data. For example, "IF FELIX IS A CAT, THEN FELIX HAS WHISKERS." If there were many cats in the problem, each would require its own statement about whiskers. In a more complex notation, the first order propositional calculus, more general statements may be made. For the example given, "IF x IS A CAT, THEN x HAS WHISKERS." This reduces the size of the initial data space, but requires considerably more complex processing to determine the applicability of a statement to a given situation. If the number of entities (e.g. cats) in a problem is small, the increase in data space size may be compensated for by the reduction in the complexity of processing, and by increased opportunities for parallelism.

Even with the data notation chosen, there remain other significant choices about how to process the data. The three primary methods are production rules (forward chaining), such as used in Mycin, backchaining, and mathematical resolution. Backchaining is the method used by the programming language PROLOG, and resolution is typically used for proving theorems in mathematical axiom systems. Each has advantages and disadvantages.

A major advantage of forward chaining is that it is relatively easy to incorporate probabilities into the data. Thus, if a given situation could have multiple causes, the relative likelihood of the alternatives can be computed. In order for this to be practical, however, the depth of the inference process and the number of alternatives must be small, otherwise the computation gets out of hand. This is the primary disadvantage of forward chaining. The process is not strongly goal-directed, and in a complex situation will (given sufficient time) deduce **everything** about the situation, in an unpredictable order. If the user's interest is focussed on only one facet of the problem, there is no way to obtain directly information about only that facet.

Backchaining starts with a completely or partially specified conclusion, and reasons backwards to check its compatibility with the data. It can be used to answer questions by using conclusions from the first order predicate calculus. For example, the conclusion "x HAS WHISKERS" would result in the substitution "x=FELIX" if the fact "FELIX IS A CAT" were added to the statement in the earlier illustration. In most cases, backchaining is faster than production rules when the desired outcome can be partially specified, because the processing is more narrowly focussed. More general questions, such as "tell me the significance of the present situation", cannot be answered by backchaining systems.

Both backward and forward (production) systems have a strong directional bias. The rules only work one way. But from a logical point of view, the rules are actually bidirectional. The rule "IF x IS A CAT, THEN x HAS WHISKERS" also implies that if x does not have whiskers, it is not a cat. Mathematical resolution is a relative of backchaining which uses a different internal notation (clause form) which captures this bidirectionality. In addition, clause form notation is simpler from an implementation standpoint than the more general formulae used in forward and back chaining. Normally, mathematical resolution is used for proof by contradiction. In this form, a hypothetical conclusion is negated, and then the data space consisting of the negated hypothesis and the known facts is examined for inconsistancy. If one exists, the original hypothesis must have been true. Resolution can also be used in the forward direction to merely generate more facts from an initial data space. In this mode, it is as undirected as production rules in forward systems.

Because of its generality and the simplicity of its implementation, we have chosen to study the *implementation of mathematical resolution as a parallel inference mechanism*. In order to keep system complexity under control at this stage of the research, we have restricted our efforts to the propositional calculus.

Mathematical resolution has a large amount of inherent parallelism, limited only by the size of the data space. Two principal parallel operations are resolvent generation and duplicate detection. Resolvent generation is the process of combining existing clauses to form new ones. This process can be carried out independently and in parallel for every pair of clauses in the data base. Of course, this requires an adequate number of processing elements and parallel access to the data. Once new clauses are generated, it is necessary to determine whether or not they already exist in the data base. Failure to delete duplicate clauses not only causes the data space to grow at an unnecessarily high rate, but it also may cause the computation to become nonterminating. The search for existing clauses which match a new one may be done in parallel by "broadcasting" the new clause into the data base and waiting for an appropriate response. Again, *parallel access to the data and a sufficient number of* computing elements are required.

The computing operations required for resolution are relatively simple boolean algebra and a small amount of counting. The necessary logic could be fabricated in either VLSI or planar integrated optics. Certainly the present state of the art in fabrication favors VLSI. However, the parallel data access requirements of the problem cannot be met in VLSI. A system performing parallel logic on a 1000 x 1000 grid would require 1,000,000 bits of information per clock cycle arriving in parallel. Pinout and crosstalk limitations in VLSI would require this information to be partially *serialized* for loading. Even if the VLSI *were* faster than the equivalent optics (and there is reason to believe that the speed of optics can be made comparable), the time required for serial data handling would limit the speed of a VLSI implementation.

There are presently a number of institutions funded to develop faster and better optical logic components. The purpose of our work is to develop a system architecture to use these components to solve reasoning problems in an effective and economical way. We intend to demonstrate the architecture using presently available optical technology insofar as possible, and to concentrate on the development of MSI level optical components using proven concepts.

RESEARCH METHOD

The approach is to first develop an optical data representation for knowledge and then to construct simulation software for all optical components to validate system behavior and reduce implementation risk. The next step is to design, simulate, and implement optical computation modules for the basic operations required for inference; optical storage and data handling modules required for data-dependent computation; and the data-dependent control required for inference. The ultimate goal is to simulate and implement a prototype optical inference engine.

As was discussed in the Rationale for Optics, the A.I. technique chosen was reasoning by resolution. In reasoning by resolution, one must begin with a set of information about the problem. Then, one takes the fact of interest and states its opposite. If the fact to be proven is true, the total set including the negation now contradicts itself. The object of resolution is to make this contradiction explicit. The technique used is to convert each fact to a binary bit string (a clause) encoding the data and systematically generate new clauses by combining existing clauses. Then, new clauses are retained which contain exactly one item of information was true in one parent and false in the other. This item is omitted in the combination and all other new clauses are discarded. The remaining new clauses are checked against all existing clauses and duplicates discarded. New clauses are generated until either an empty clause is generated or no new clauses can be formed. If the empty clause is produced, the original fact was true, otherwise it was false.

The optical inference system is designed around medium scale optical components. Major components under study and development are:

- o Parallel read - parallel write memory storage frames which will allow parallel access to entire matrices of stored data.
- o Switchable prism multiplexors and shifters which will allow spatial switching of bit vector and matrix data.
- o Optical pushdown stacks which are a combination of the memory frames and the prism shifters. These will allow vector-at-a-time access to data matrices.
- o Resolver (clause combiner) which will use medium scale boolean logic to produce a matrix of clauses from an old matrix and a new clause.
- o Duplicate detector which will compare a new clause against the existing clause base in parallel to rapidly reject duplicates.

RESULTS

A system was designed and simulated. Several components were constructed and tested. Some of the components are acceptable for incorporation into a complete system that is reduced in size. Some, however, are marginally acceptable and more research is needed to find better approaches.

Data representations were selected and analyzed, the system was designed and functional simulation was performed. Phenomenological simulations for the key elements of the system were performed. Two versions of a prototype gate have been built and tested and a Proof-of-Concept 2:1 multiplexor was built and tested. A dual bit photoaddressed FLC memory element has been built and tested.

The system was functionally simulated, varying several system design parameters. In approximate order of importance, these parameters are:

1) Parallel versus serial frame compaction. In parallel compaction, valid resolvents are immediately squeezed to the top of the frame. In serial compaction, the whole frame must be shifted and checked. This has a major impact on performance.

2) Generation storage strategy. In separated storage, data generated in each pass through the system is kept separate from previous data. This takes more storage, but reduces computing. In compacted storage, frames are added to until full.

3) Extent of duplicate checking. In complete checking, duplicates within the same pass (generation) are detected. In partial checking, only older duplicates are found.

4) Parallel duplicate checking. In parallel checking, the entire data store is checked in one operation. In serial checking, one frame of the data store is checked at a time. This affects the performance if the faster options of the preceding choices are selected.

Table 1. gives the results of that simulation. The particular problem chosen is given in Appendix I.

The data representation chosen was the positional dibit notation for clauses (the basic unit of knowledge in mathematical resolution). [1986 SPIE Conference on Optical Computing] This is illustrated in Figure 1.

The fundamental data element in a clause is the literal, which can occur in asserted form (X) or negated form (X'), or simply be omitted. Each row in the notation corresponds to a clause. Each pair of columns in the notation corresponds to a literal, with the bit pair 01 used for assertion, 10 used for negation, and 00 for omission. Each bit is encoded in polarized light as shown in Figure 1

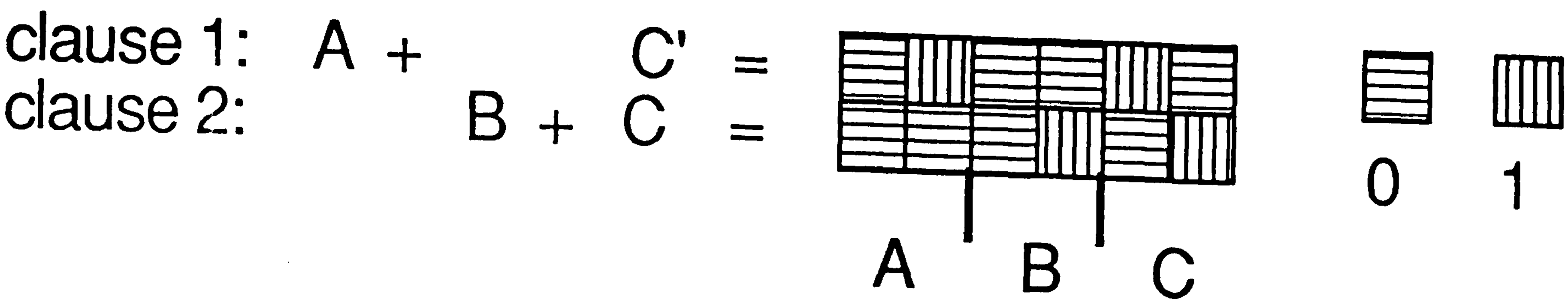


FIGURE 1, DIBIT NOTATION

problem size: 15 rules, 19 clauses, 31 literals

ALGORITHM: 'SET-OF-SUPPORT - Serial timing 35,609 cycles

FRAME COMPACTION STRATEGY:

DUPLICATE DETECTION STRATEGY ---	Serial		Parallel	
	separated	compacted	separated	compacted
complete check - parallel	62,290 (175%)	41,289 (116%)	676 (1.9%)	1,682 (4.7%)
newest generation not checked - parallel	136,911 (384%)	20,020 (56%)	1,182 (3.3%)	1,128 (3.2%)
complete check - frame-by-frame	62,865 (176%)	41,533 (116%)	1,251 (3.5%)	1,926 (5.4%)
newest generation not checked- fran.e-by-frame	138,295 (388%)	20,452 (57%)	2,566 (7.2%)	1,560 (4.4%)

GENERATION STORAGE STRATEGY

TABLE 1 SIMULATION RESULTS

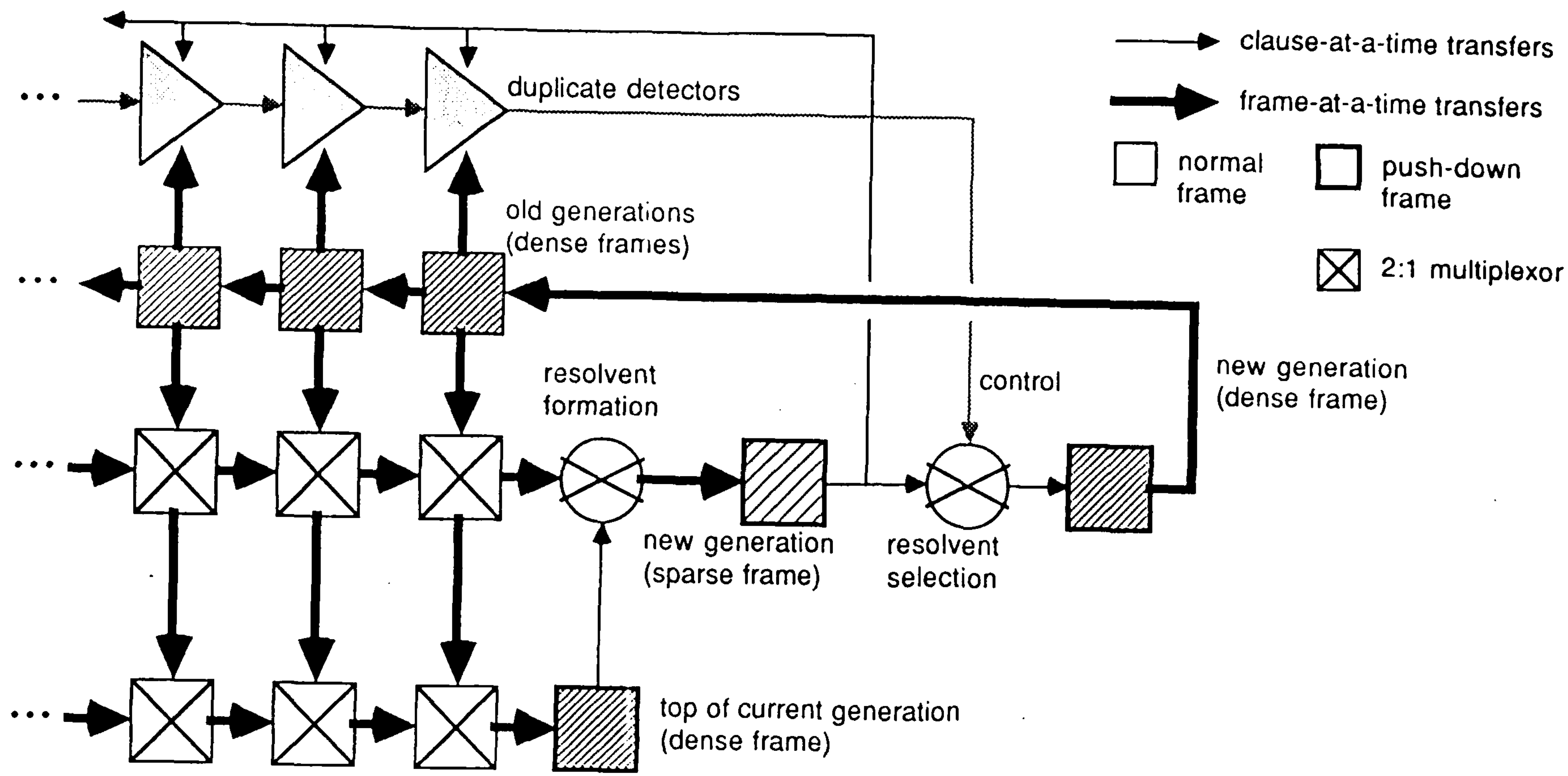


FIGURE 2, OPTICAL RESOLUTION SYSTEM

The data handling and storage used a functional and data flow design of a data compaction system and was described in Applied Optics, May 1987. A review of plausible optical data storage techniques was given at the 1987 Computers in Aerospace Conference. Because data-dependent control is needed, a review of control techniques for mathematical resolution was performed. The set-of-support strategy was selected as offering the best balance between efficiency and complexity of optical implementation.

The optical resolution system is shown in Figure 2. In this figure, clauses are stored in the "old generation" push down stack. Frame-at-a-time multiplexors are used to route current information to the resolvent formation logic, which systematically combines a single clause with an existing frame to produce a frame of potential new resolvents. These are checked for validity and for duplication of existing data, and useful new resolvents are placed in the new generation. As resolution proceeds, new generation data is added to the old generations to allow the process to proceed. Figure 3 shows the operation of duplicate detection.

Ferroelectric liquid crystals are key to many of the components of the system. In one application, the fast low-loss polarization rotation characteristics are used and in others, their potential as a high-speed switch is employed. Figure 4 illustrates both of these uses.

Figure 5 shows how the switching of the index of refraction can be employed to fabricate a position shifter. This concept has been demonstrated and a shifter is being fabricated. A shifter of this sort could be used to effect compaction of sparse matrices into dense matrices as shown in Figure 6.

A logic gate has been designed that uses polarization logic. It is illustrated in Figure 7, where the possible inputs are the four combinations of polarized light shown at station "A". The outputs at station "D" implement the "or" or "and" boolean logic function depending on whether logic 1 is chosen as horizontal or vertical polarization.

An experimental model was constructed as shown in Figure 8 and its robust operation demonstrated. This component also was simulated and the comparison of the results was presented at the 1987 International Commission on Optics.

A parallel storage device with photoaddressed spatial light modulators (SLMs) has been designed and construction of the photoaddressed SLM has started. The storage device is shown in Figure 9.

FUTURE RESEARCH

The current logic gate is too complex for economical array-level integration. We need improved fabrication methods or simpler gate design. The signal quality from real electro-optic components is limited, requiring polarization and/or amplitude restoration frequently. The fan-in requirements of some of the system components cannot be accommodated without excessive component cost. The average performance level of FLC's must be improved to approach optimum values. The FLC switchable prism contrast ratio is limited, requiring thresholding and signal restoration but recent results indicate that a much higher contrast is possible. The true relative importance of key system design parameters is not known over the range of problem sizes and structures of interest.

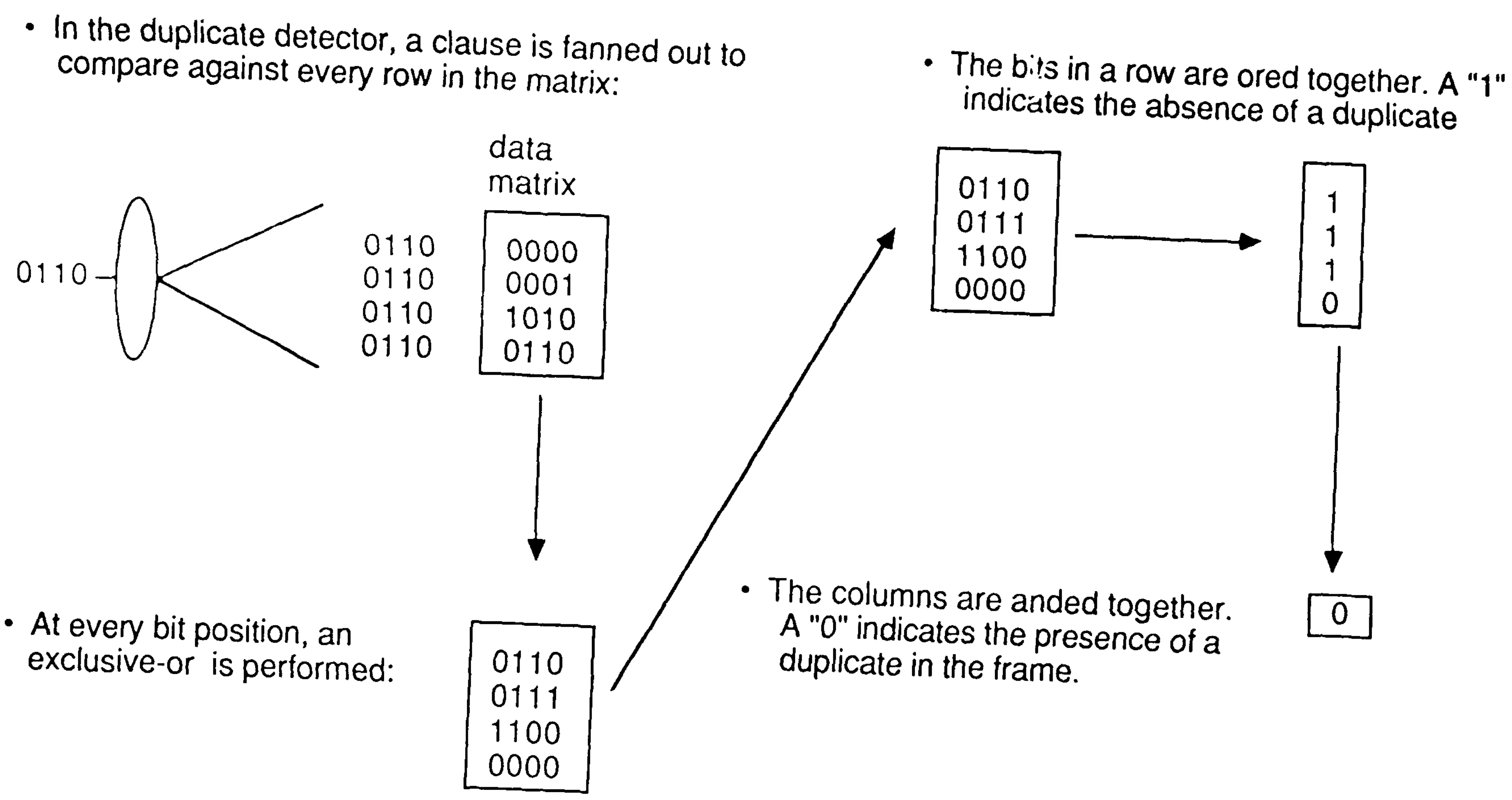


FIGURE 3, DUPLICATE DETECTION

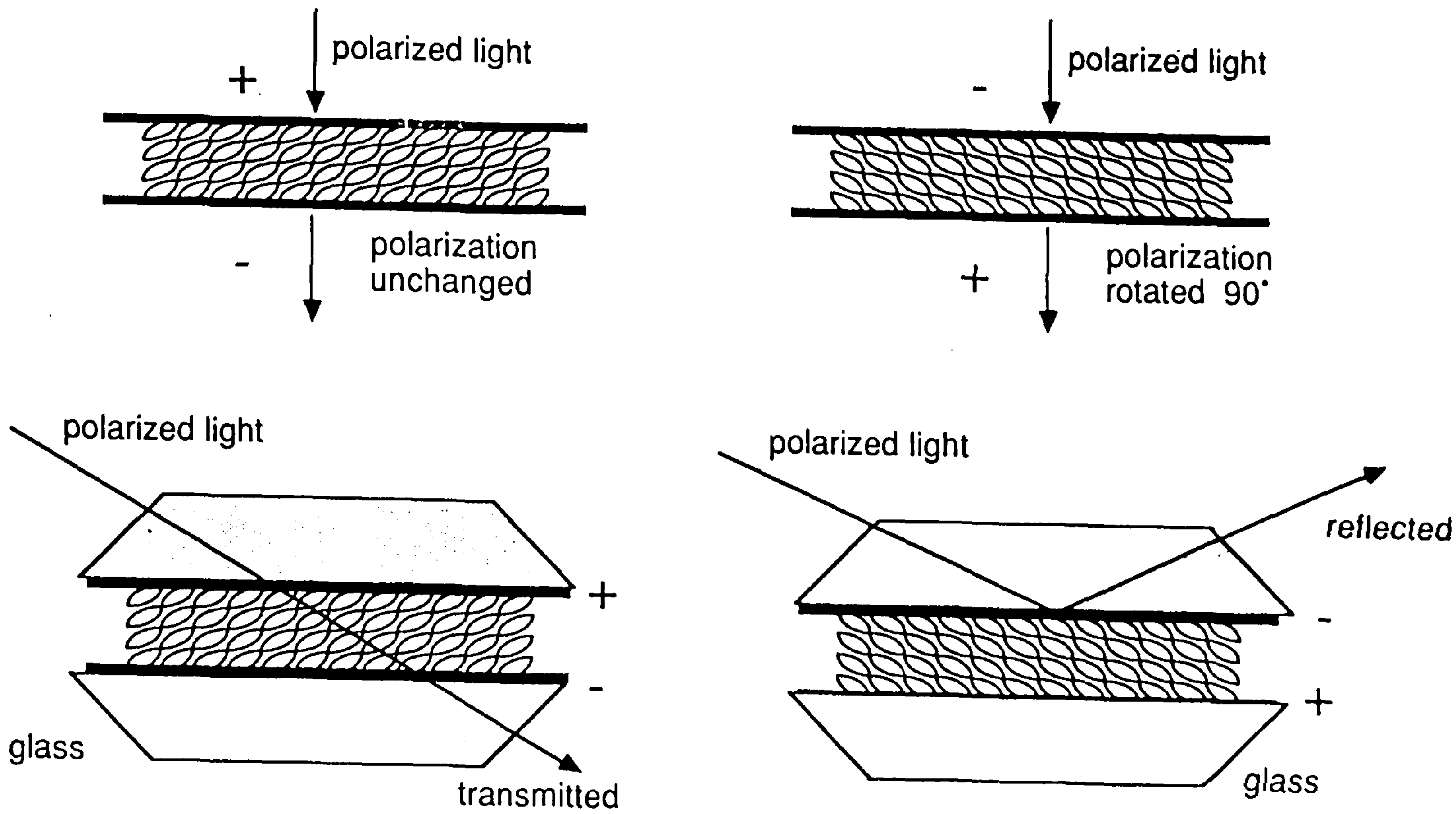


FIGURE 4, FERROELECTRIC LIQUID CRYSTALS

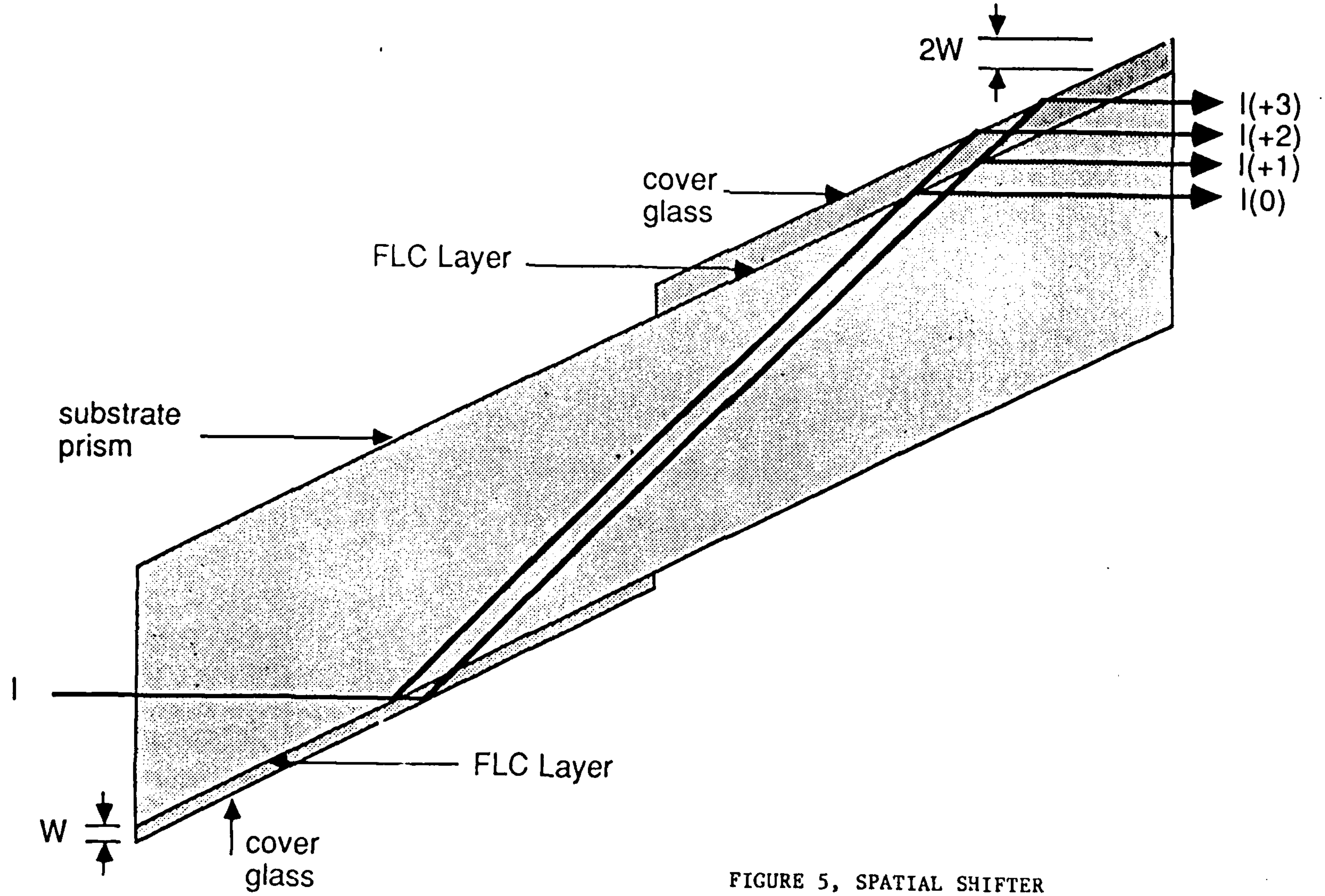


FIGURE 5, SPATIAL SHIFTER

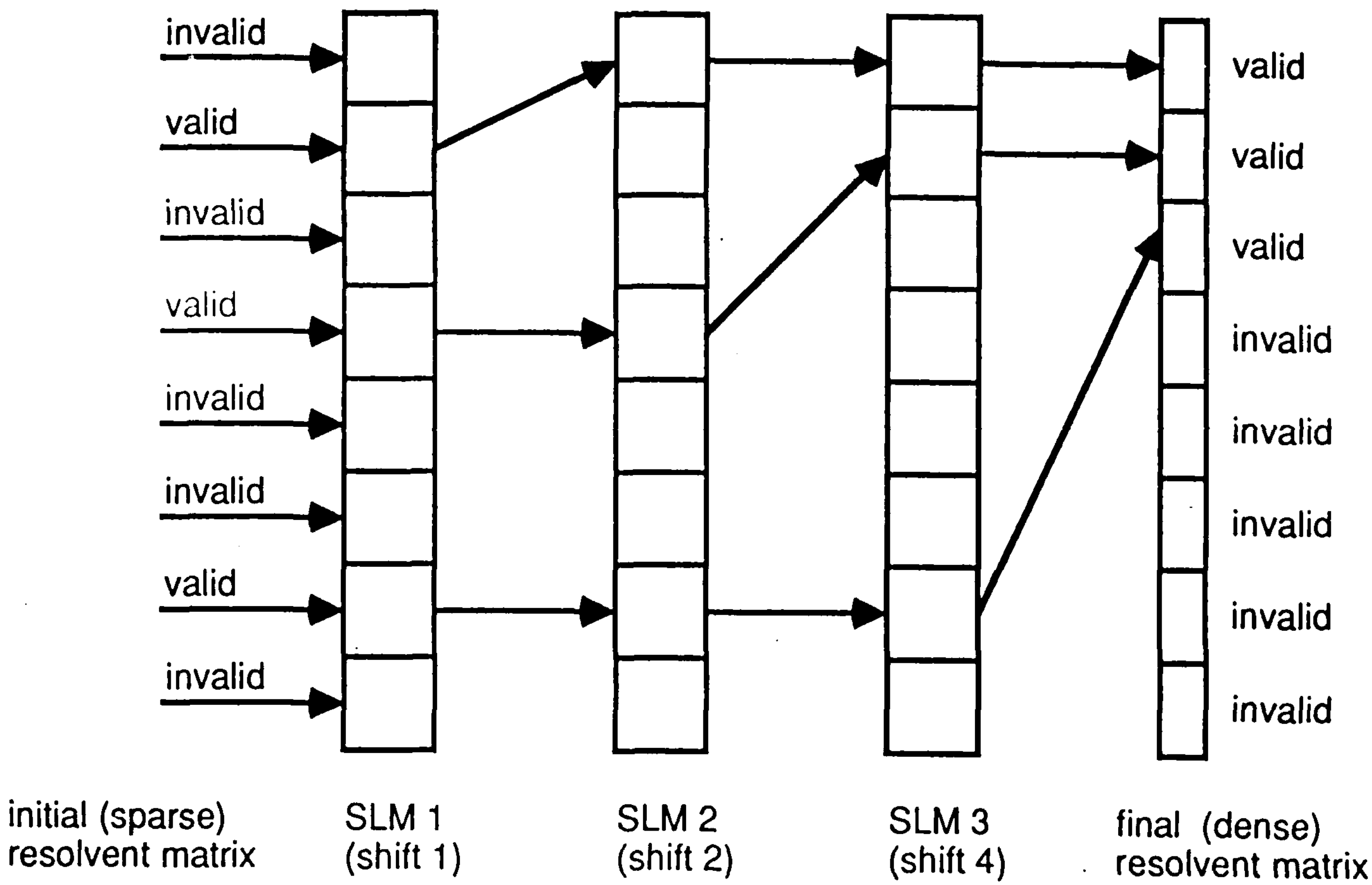


FIGURE 6, COMPACTION LOGIC

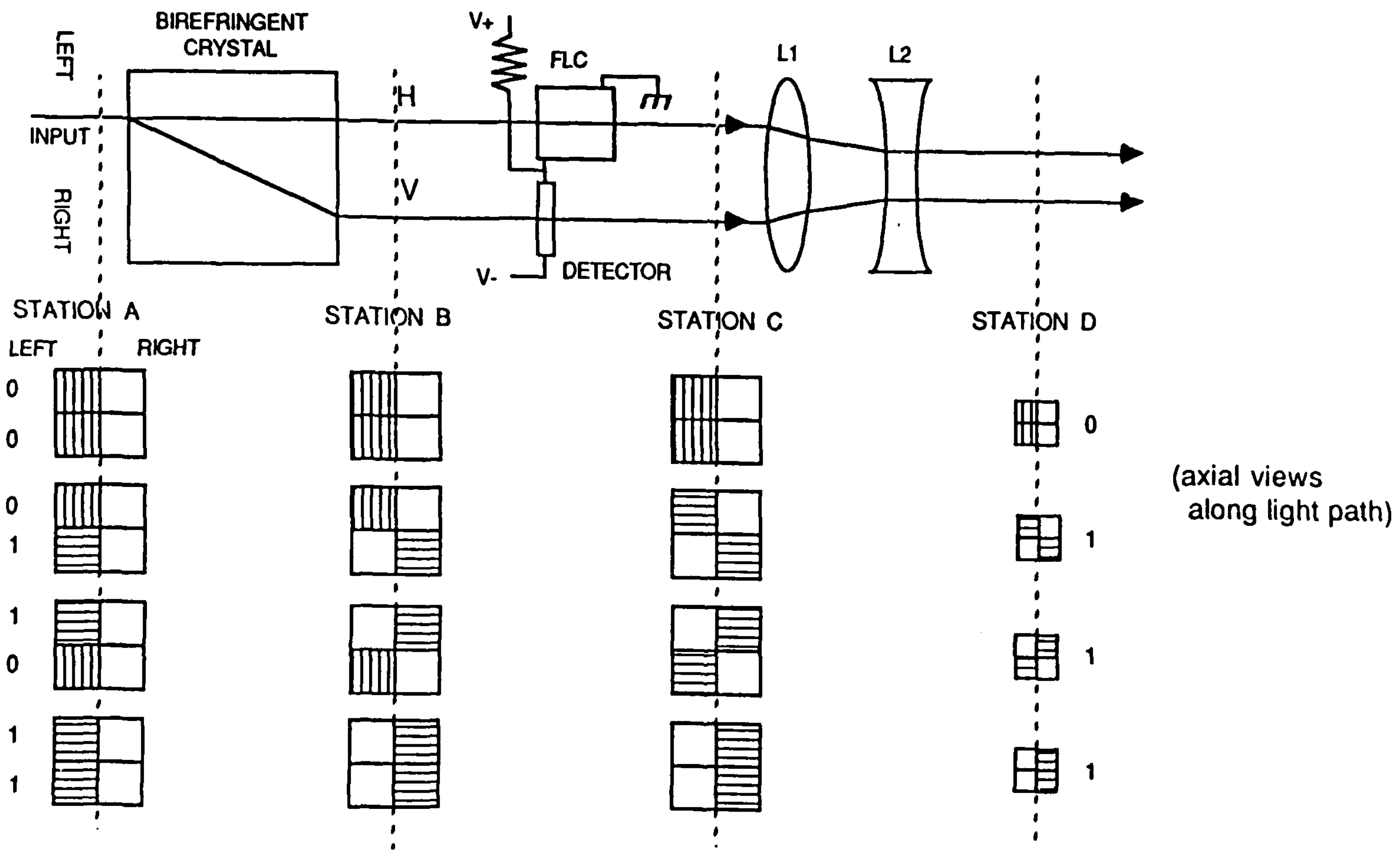


FIGURE 7, POLARIZED LIGHT LOGIC GATE

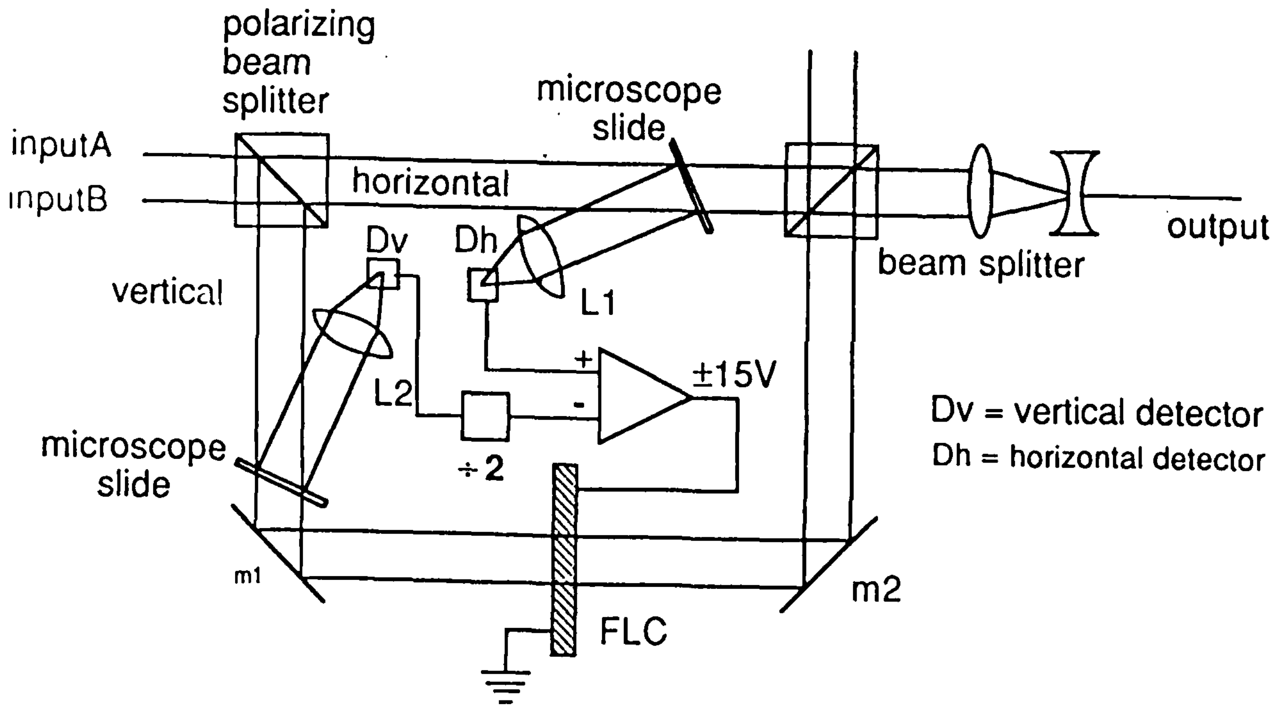


FIGURE 8, PROTOTYPE LOGIC GATE

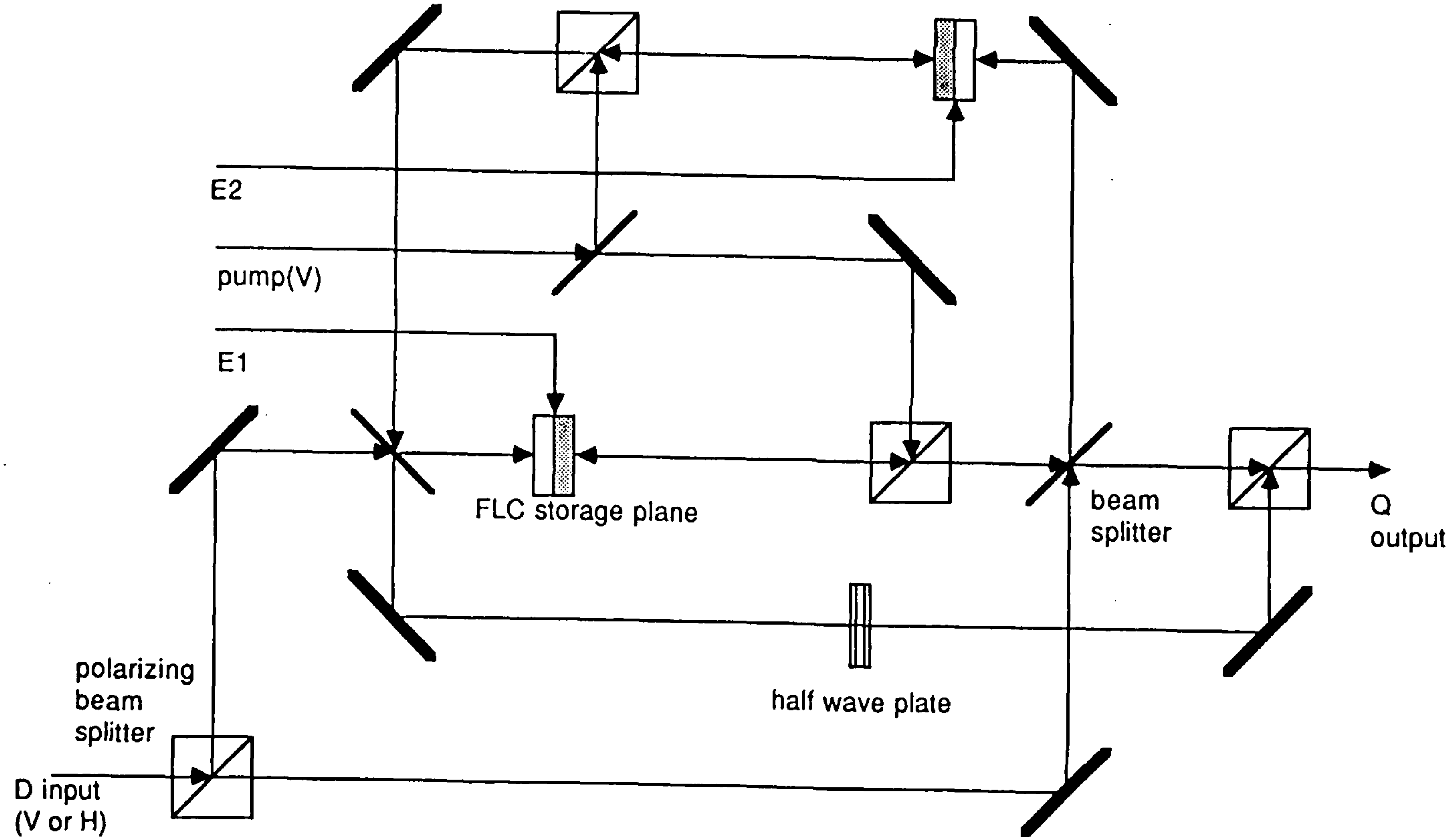


FIGURE 9, FLC MEMORY FRAME

We believe that many of the FLC related issues are connected with our use of components not specifically designed for our purposes. Issues such as the refractive index of cover glass, conductive coating thickness, liquid crystal material and surface treatment are all more critical in this application than in on/off visual displays, which is what governs many current FLC designs.

Research on the architecture and the components to address these issues is planned.

PUBLICATIONS

"Optical Representations for Artificial Intelligence Problems", R.A. Schmidt & W.T. Cathey, SPIE Proceedings, Vol. 625, January 1986 (to appear June 1988 in Optical Engineering).

"Photoaddressing of High Speed Liquid Crystal Spatial Light Modulators", G. Moddel, K.M. Johnson & M.A. Handschy, SPIE Proceedings, Vol. 754, 1987.

"Optical Implementations of Mathematical Resolution", R.A. Schmidt & W.T. Cathey, Applied Optics, Vol. 26, May 15, 1987.

"Low Loss Polarization Based Optical Logic Gates", W.T. Cathey & R.A. Schmidt, 14th Congress of the International Commission for Optics, August 24, 1987, Quebec, Canada, (SPIE Proceedings, Vol. 813).

"Polarization-Based Optical Parallel Logic Gate Utilizing Ferroelectric Liquid Crystals", M.A. Handschy, K.M. Johnson, W.T. Cathey, and L.A. Pagano-Stauffer, Ptics Letters Vol. 12, August 1987.

"Memory Systems for Optical Computing", R.A. Schmidt and G. Moddel, AIAA Computers in Aerospace VI Conference (AIAA, Washington, D.C., 1987), pp. 201-206.

"Design and Performance of High-Speed Photoaddressed Spatial Light Modulators", W.Li, C.T. Kuo, G. Moddel and K.M. Johnson, Proc. SPIE 936 (1988).

"Optical Addressing of High-Speed Spatial Light Modulators with a-Si:H", G. Moddel, C.T. Kuo, K.M. Johnson and W. Li, Amophous Silicon Technology, (Materials Research Society, Pittsburgh, 1988).

APPENDIX I

A REASONING PROBLEM

(From Winston, Artificial Intelligence, p. 177)

1) If the animal has hair then it is a mammal.
2) If the animal gives milk then it is a mammal.
3) If the animal has feathers then it is a bird.
4) If the animal flies and it lays eggs then it is a bird.
5) If the animal is a mammal and it eats meat then it is a carnivore.
6) If the animal is a mammal and it has pointed teeth and it has claws and its eyes point forward then it is a carnivore.
7) If the animal is a mammal and it has hoofs then it is an ungulate.
8) If the animal is a mammal and it chews cud then it is an ungulate and it is even-toed.
9) If the animal is a carnivore and it has a tawny color and it has dark spots then it is a cheetah.
10) If the animal is a carnivore and it has a tawny color and it has black stripes then it is a tiger.
11) If the animal is an ungulate and it has long legs and it has a long neck and it has a tawny color and it has dark spots then it is a giraffe.
12) If the animal is an ungulate and it has a white color and it has black stripes then it is a zebra.
13) If the animal is a bird and it does not fly and it has long legs and it has a long neck and it is black and white then it is an ostrich.
14) If the animal is a bird and it does not fly and it swims and it is black and white then it is a penguin.
15) If the animal is a bird and it flies then it is an albatross.

PROVE:

- If the animal has hair and it has hoofs and its color is white and it has black stripes then it is a zebra.
- If the animal has a tawny color and it has dark spots and it gives milk and it chews cud and it has a long neck and it has long legs, then it is a giraffe.

These rules are translated into a Lisp input as:

(hairy implies mammal)
(lactates implies mammal)
(feathers implies bird)
((flies and (lays eggs)) implies bird)
((mammal and (eats meat)) implies carnivore)
((mammal and (pointed teeth) and claws and (looks forward)) implies carnivore)
((mammal and hoofs) implies ungulate)
((mammal and (chews cud)) implies (ungulate and (even toed))
((carnivore and tawny and (dark spots)) implies cheetah)
((carnivore and tawny and (black stripes)) implies tiger)
((ungulate and (long legs) and (long neck) and tawny and (dark spots)) implies giraffe)
((Ungulate and white and (black stripes)) implies zebra)
((bird and (not flies) and (long legs) and (long neck) and (black and white)) implies ostrich)
((bird and (not flies) and swims and (black and white)) implies penguin)
((bird and flies) implies albatross)
*
((hairy and hoofs and white and (black stripes)) implies zebra

In clause form this becomes:

1' + 2
3' + 2
4' + 5
6' + 7' + 5
2' + 8' + 9
2' + 10' + 11' + 12' + 9
2' + 13' + 14
2' + 15' + 14
2' + 15' + 16
9' + 17' + 18' + 19
9' + 17' + 20' + 21
14' + 22' + 23' + 17' + 18' + 24
14' + 25' + 20' + 26
5' + 6 + 22' + 23' + 27' + 28
5' + 6 + 29' + 27' + 30
5' + 6' + 31
*
1
13
25
20
26'

END
DATE
FILMED
9-88
DTIC